REMARKS

Claims 1-2, 4-6, 12-13, 15-18, 21-24, 28-30, 32, 34-40, 42-49, 51-58, and 60-78 were presented for examination, of which claims 1, 5, 12, 17, 24, 32, 34, 43, 52, 61, 66, 70, and 74 are independent. Claims 3, 7-11, 14, 19-20, 25-27, 31, 33, 41, 50 and 59 were previously canceled. Applicants amend claims 1, 5, 12, 17, 24, 32, 34, 43, 52, 61, 66, 70, and 74 herein. No new matter is added.

Applicants thank the Examiner for withdrawing the rejections under 35 U.S.C. §112 and §103 (Office Action at page 2).

I. Finality of Office Action

Applicants note for the record that, at page 1, the Office Action is identified as a final Office Action. However, at page 2, the Office Action acknowledges the Request for Continued Examination filed on December 5, 2008. Accordingly, the Office Action should properly be a non-final Office Action. Further, the Office Action is identified as non-final in PAIR.

Applicants contacted the Examiner by telephone on February 23, 2009. At that time, the Examiner confirmed that the current Office Action is non-final.

II. Claim Rejections under 35 U.S.C. §103

Claims 1-2, 4-6, 12-13, 15-18, 21-24, 28-30, 32, 34-40, 42-49, 51-58, and 60-78 have been rejected under 35 U.S.C. §103(a) as being anticipated by United States Patent Publication No. 2002/0083413 to Kodosky et al. (hereafter "Kodosky") in view of *Using Real Time Expert Systems for Control System Prototyping* by Arzen (hereafter "Arzen"). Applicants respectfully traverse this rejection.

A. Claims 1-2, 4, and 6

Applicants respectfully submit that Kodosky and Arzen, alone or in any reasonable combination, do not disclose or suggest *representing the function graphically ... in the graphical representation of the finite state machine*, nor that *the function that is represented graphically has a function prototype that specifies a syntax for invoking the function, the*

function prototype specifying a function name for the function, and the function is defined in the statechart system in a graphical language, which is present in independent claim 1.

The present Application is generally directed to graphical representations of functions that may be described, defined, represented, or invoked in a modeling system for finite state machines. As noted in the Application at page 1, "existing statechart systems for modeling finite state machines permit a user to embed textual definitions of functions in a statechart and invoke those functions in the statechart." The present Application allows a user to "represent visually a procedure performed by a function in a statechart system." (Application at page 3, emphasis added).

Textually-defined functions in a statechart can be difficult to define and edit. In the textually-defined functions of statechart systems that existed at the time the invention was made, the procedure performed by the function was defined by external code (Specification at page 1). In order to textually define a function, a user would need to learn the language of the textual code that is external to the statechart system. Textually defined functions referencing external code utilized debuggers and parsers that were external to the statechart system. Using external debuggers and parsers requires more system resources and makes representing the code in the statechart more difficult and cumbersome.

On the other hand, providing graphically-represented functions in a statechart environment, as recited in claim 1, realizes benefits that are not possible with externally-defined code. When a graphically-represented function is provided *in* a statechart environment (rather than being provided outside the statechart environment), features of the statechart environment can be leveraged for use with the graphically represented function. For example, when a graphically-represented function is embedded in a statechart environment, the statechart environment's graphical parsing capabilities can be leveraged to check the diagram, including the graphically defined function, for errors. Error checking of textually-defined functions used with statechart environments is done with an external error checker, which is more complicated and requires more system resources.

Further, a statechart's animation and debugging capabilities can be used to graphically step through the graphical function, allowing the function to be debugged without external

debugging tools and without the need to separately generate code for the function. Still further, instead of referencing textual code to define the procedures carried out by the function, a graphically defined function allows a user to use a *diagram* to represent the procedure performed by the function. This improves readability and makes the graphical function easier to generate and modify (Specification at page 3).

Kodosky is one example of an environment that utilizes externally-defined code (see, e.g., Kodosky at paragraphs [0168]-[0169]). Kodosky is generally directed to a system and method for generating a graphical program in response to state diagram information (Kodosky at Abstract). Kodosky includes a state diagram that specifies a plurality of states and state transitions. A graphical program generator generates a framework for source code that includes placeholders a user fills in with code for states and transitions (Kodosky at Abstract).

Kodosky does not address graphically represented functions like the functions recited in claim 1. For example, the Examiner recognizes that Kodosky does not disclose that *the function that is represented graphically has a function prototype that specifies a syntax for invoking the function, the function prototype specifying a function name for the function* (Office Action at page 4). Instead, the Examiner relies on Arzen for the above-quoted feature of claim 1.

Arzen is generally directed to real-time expert systems developed from rule-based systems. As part of the development process, Arzen uses graphical languages that include blocks. In the passage of Arzen cited by the Examiner, Arzen states that "a function block can be seen as an instance of a class that describes the connection terminals, attributes, function, and graphical representation of the block."

As noted above, graphical functions in state diagram environments can leverage the state diagram environment's capabilities, such as animation, debugging, and parsing capabilities. Using the state diagram environment for these tasks conserves system resources because it does not require the use of outside debuggers or parsers. This is not possible in either Arzen or Kodosky.

One indicia of nonobviousness is that the elements in combination do not merely perform the function that each element performs separately (MPEP at §2141.V). Providing graphical

functions for use with finite state machines allows for synergies that are not present when graphical functions and finite state machines are considered separately.

In order to better clarify these aspects of claim 1, Applicants amend claim 1 to recite that *the function is defined in the statechart system in a graphical language*. Applicants respectfully submit that Kodosky and Arzen, alone or in any reasonable combination, do not disclose or suggest a function that is defined in the statechart system. As discussed above, Kodosky references externally defined code, while Arzen does not define graphical functions in the statechart system.

Further, claim 1 recites that *the function is graphically represented ... in the graphical representation of the finite state machine*, and that *the function that is represented graphically has a function prototype that specifies a syntax for invoking the function*. An example of such a graphically represented function is depicted in Figure 2. By providing the graphically represented function with a function prototype, the statechart environment's native parser can be used to parse the graphical function. Accordingly, it is not necessary to use an external parser to parse the graphical function. Neither Kodosky nor Arzen disclose or suggest these features of claim 1. The Examiner recognizes that Kodosky does not disclose a function prototype (Office Action at page 4). The Examiner cites the "function block" on page 25 of Arzen as an example of a function prototype (Office Action at page 4). However, Arzen defines a function block as an instance of a class that describes the function of the block (Arzen at page 25, last full paragraph). The "function block" of Arzen is an instance of a class, and not a function with a function prototype. There is no discussion in Arzen of a function prototype that specfies a syntax for invoking the function.

Still further, one having ordinary skill in the art would not modify Kodosky so that *function is defined in the statechart system in a graphical language*, as recited in claim 1. Kodosky is directed to a system that takes a pre-existing state diagram and generates a graphical program generation program (GPG program) from the pre-existing state diagram. Thus, Kodosky is not concerned with the way that functions are defined in the statechart system. Instead, Kodosky is concerned with the product of the statechart system (the state diagram).

Accordingly, there is no motivation to modify Kodosky to change the way that functions are defined in a statechart system.

In light of the above, Applicants respectfully submit that Kodosky and Arzen, alone or in any reasonable combination, do not disclose or suggest the features of claim 1. Claims 2, 4, and 6 depend from claim 1, and therefore include each and every element of claim 1. Applicants respectfully request that the Examiner reconsider and withdraw the 35 U.S.C. §103(a) rejection of claims 1-2, 4, and 6.

B. Claim 5

Independent claim 5 recites that *the function is represented graphically in the statechart system as a diagram comprising graphical elements and has a function prototype that specifies a syntax for invoking the function, the function prototype specifying a function name for the function*. The Examiner recognizes that Kodosky does not disclose or suggest this element of claim 5 (Office Action at page 5). Instead, the Examiner relies on Arzen for a function prototype.

Applicants amend claim 5 to better claim the invention. Specifically, Applicants amend claim 5 to recite that *the function is represented graphically in the statechart system as a diagram*. As noted above with respect to claim 1, neither Kodosky nor Arzen discloses graphical functions in a statechart system.

Further, as noted above with respect to claim 1, simply combining Arzen and Kodosky does not result in the graphical functions of claim 1. By providing the graphically represented function with a function prototype, as in claim 5, the statechart environment's functionality can be leveraged in ways not apparent from Kodosky and Arzen. Fo example, the statechart environment can be used to parse, debug, and error-check the function. A combination of Kodosky and Arzen would not result in a graphically represented function that *and has a function prototype that specifies a syntax for invoking the function, the function prototype specifying a function name for the function*.

In light of the above, Applicants respectfully submit that Kodosky and Arzen, alone or in any reasonable combination, do not disclose or suggest the features of claim 5. Applicants respectfully request that the Examiner reconsider and withdraw the 35 U.S.C. §103(a) rejection of claim 5.

C. Claims 12-13, 15-18, 21-23, 24, 28-30, 32, 34-40, 42-49, 51-58, and 60-78

1. Claims 12-13, 15-16, 24, 28-30, 32, 43-49 and 51

Amended independent claim 12 recites *the graphical function having a function prototype in the statechart system that specifies a syntax for invoking the graphical function, the function prototype specifying a function name for the graphical function.*

Amended independent claims 24 and 32 recite *a function prototype represented in the statechart system.*

Amended independent claim 43 recites *the graphical function having a function prototype in the event-driven system environment.*

As noted above with respect to claims 1 and 5, Arzen and Kodosky, alone or in any reasonable combination, do not disclose or suggest a function prototype in a statechart or event-driven system environment.

Claims 13 and 15-16 depend from claim 12, and therefore include each and every element of claim 12. Claims 28-30 depend from claim 24, and therefore include each and every element of claim 24. Claims 44-49 and 51 depend from claim 43, and therefore include each and every element of claim 43.

2. Claims 17-18 and 21-23

Amended independent claim 17 recites *means to represent the graphical function as an executable state flow diagram in a statechart system.* In light of the above remarks, Applicants respectfully submit that Kodosky and Arzen, alone or in any reasonable combination, do not represent the graphical function as an executable state flow diagram in a statechart system.

Claims 18 and 21-23 depend from claim 17, and therefore include each and every element of claim 17.

3. Claims 34-40 and 42

Amended independent claim 34 recites that *said function comprises at least two graphical components in the event driven-system modeling environment*. In light of the above remarks, Applicants respectfully submit that Kodosky and Arzen, alone or in any reasonable combination, do not disclose or suggest a function comprising at least two graphical components in an event-driven system modeling environment. Claims 35-40 and 42 depend from claim 34, and therefore include each and every element of claim 34.

4. Claims 52-58 and 61-65

Amended independent claim 52 recites *a graphical function defined in the event-driven system environment*.

Independent claim 61 recites *graphically representing a function defined in the graphical block diagram environment*.

In light of the above remarks, Applicants respectfully submit that Kodosky and Arzen, alone or in any reasonable combination, do not disclose or suggest a graphical function defined in the event-driven system environment.

Claims 53-58 and 60 depend from claim 52, and therefore include each and every element of claim 52. Claims 62-65 depend from claim 61, and therefore include each and every element of claim 61.

5. Claims 66-73

Amended independent claim 66 recites *graphically defining a function in the graphical block diagram environment*.

Independent claim 70 recites *a function defined in the graphical block diagram environment*.

In light of the above remarks, Applicants respectfully submit that Kodosky and Arzen, alone or in any reasonable combination, do not disclose or suggest graphically defining a function in a graphical block diagram environment.

Claims 67-69 depend from claim 66, and therefore include each and every element of claim 66. Claims 71-73 depend from claim 70, and therefore include each and every element of claim 70.

6. Claims 74-78

Amended independent claim 74 recites that *at least a subset of commands of the graphical function are defined through a graphical representation in the graphical block diagram modeling environment*. In light of the above remarks, Applicants respectfully submit that Kodosky and Arzen, alone or in any reasonable combination, do not disclose or suggest defining at least a subset of commands of the graphical function through a graphical representation in the graphical block diagram modeling environment. Claims 75-78 depend from claim 74, and therefore include each and every element of claim 74.

In light of the above, Applicants respectfully request that the Examiner reconsider and withdraw the 35 U.S.C. §103(a) rejection of claims 12-13, 15-18, 21-23, 24, 28-30, 32, 34-40, 42-49, 51-58, and 60-78.

CONCLUSION

In view of the above, Applicants believe the pending application is in condition for allowance and urge the Examiner to pass the claims to allowance. Should the Examiner feel that a teleconference would expedite the prosecution of this Application, the Examiner is urged to contact the Applicants' attorney at (617) 227-7400.

Please charge any shortage or credit any overpayment of fees to our Deposit Account No. 12-0080 under Order No. MWS-070RCE3. In the event that a petition for an extension of time is required to be submitted herewith, and the requisite petition does not accompany this response, the undersigned thereby petitions under 37 C.F.R. §1.136(a) for an extension of time for as many months as are required to render this submission timely. Any fee due is authorized to be charged to the aforementioned Deposit Account.

Dated: April 21, 2009

Respectfully submitted,

Electronic signature: /John S. Curran/
John S. Curran
Registration No.: 50,445
LAHIVE & COCKFIELD, LLP
One Post Office Square
Boston, Massachusetts 02109-2127
(617) 227-7400
(617) 742-4214 (Fax)
Attorney/Agent For Applicant